

# Formation And Maintenance of Self-Organizing Wireless Networks

Keith Scott  
Keith.Scott @jpl.nasa.gov  
Jet Propulsion Laboratory  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099

Nicholas Bambos  
bambos@leland.stanford.edu  
Department of ???  
Stanford University  
Stanford, CA

## Abstract

*There are numerous military, commercial, and scientific applications for mobile wireless networks which are able to self-organize without recourse to any pre-existing infrastructure. We present the Self-Organizing Wireless Adaptive Network (SWAN) protocol, a distributed networking protocol capable of managing such networks. The SWAN approach is based on **dynamic topology management with power control**, allowing it to adapt gradually to the changing environment instead of periodically discarding the network topology information and rebuilding the network from scratch. In addition, under SWAN **control information is distributed** instead of being concentrated in a "control phase". This provides significant savings when the acquisition times of the modems are high.*

## 1 Introduction

Given a set of mobile nodes or terminals equipped with radio transceivers, we would like for the nodes to be able to "self-organize" into a wireless network. That is, the nodes must cooperate to form a multi-hop network amongst themselves without recourse to any pre-existing infrastructure. Since the nodes are mobile, the network must continually adapt to their changing positions.

A second major challenge in such a network is to ensure that the transmission schedules of the various nodes are compatible. If a pair of nodes is using a particular transmission resource, whether it is a narrow-band frequency slot, a time slot, or a spread-spectrum code, that resource will be unavailable to other nearby

users. For example, if the nodes in a wireless network are using direct sequence spread spectrum radios to communicate, there is the danger that one transmitter may completely dominate several receivers in his vicinity, preventing them from receiving from any other node in the network (the near-far problem).

The rest of this paper is organized as follows: XXX  
YYY ZZZ.

## 2 Background

### 2.1 Other ad-hoc network methods

Several authors have examined protocols for self-organizing networks [FSM89, BFJ31, PKS85, BR90, RS86, GT95]. A common approach to maintaining the network in the face of node mobility and changing radio conditions is to periodically tear down the entire network structure and to regenerate the network topology from scratch. These methods continually cycle through alternating phases of network operation. In the first phase, the protocols gather information about the network topology. This information is then used to form a compatible transmission schedule for the second phase.

This cyclic approach to network management has two main disadvantages. First, forcing all of the nodes to periodically participate in the topology gathering phase induces a large and possibly unnecessary amount of overhead. This becomes particularly difficult as the acquisition time (the time required for a receiver to "lock on" to a transmitter's signal for the first time) increases. Second, generating a compatible transmission schedule is a complex task, especially as

the network size grows.

## 2.2 Distributed Power Control

SWAN uses the Distributed Power Control with Active Link Protection (DPC-ALP) power control algorithm [CBP94] to ensure that ongoing transmissions are not accidentally destroyed by new ones. Under DPC-ALP, transmitter powers are updated in a series of steps, and every transmission is in one of two states, active or inactive. The transmitter power at step  $i+1$  is a function of the transmit power in step  $i$ , the transmission state at step  $i$ , the desired signal to noise ratio  $\gamma$ , and the received SNR at step  $i$  (fed back from the receiver to the transmitter). Thus if there are  $N$  transmissions, with the power of the  $i^{th}$  transmission during the  $k^{th}$  step given by  $P_i^k$  then:

$$P_i^{k+1} = \begin{cases} \delta P_i^k & \text{Inactive Transmissions} \\ \delta P_i^k \frac{\gamma}{\text{SNR}_i^k} & \text{Active Transmissions} \end{cases} \quad 1 \leq i \leq N \quad (1)$$

All transmissions begin in the inactive state at a very low power (possibly commensurate with the noise power as seen by a typical receiver). A transmission becomes active once its received SNR crosses the threshold  $\gamma$ . A consequence of DPC-ALP is that, in a static network, if a transmission becomes active at step  $k$ , it will remain active for all  $t > k$  regardless of the number and locations of new transmissions started after time  $k$ . This is because active transmissions are more aggressive in updating their powers, and may in fact prevent other inactive transmissions from becoming active.

## 3 The SWAN protocol

As with many other ad-hoc networking methods, SWAN divides time into a repeating series of frames, which are further subdivided into slots, as shown in figure 1. SWAN employs the DPC-ALP power control algorithm during most of these slots to ensure that ongoing transmissions are not interrupted by transmissions being set up. In addition, there is a single slot at the beginning of each frame (possibly of a different size) that is not subject to any power control restrictions. This random access period is used to

allow nodes to form new connections within the network. Once a connection has been established between two nodes, they establish control calls to exchange the data needed to maintain the power control algorithm. These control calls form a natural means for distributing information such as routing tables.



Figure 1: The TDMA frame, divided into a short random-access period and a data subframe.

### 3.1 link formation and topology maintenance

To form new links in the network, nodes handshake during the random access periods at the beginning of each TDMA frame. Each stage of the handshaking takes place during a *different* frame, so that if node  $i$  looks for neighbors during frame  $k$ , a node may respond to him during the RA period in frame  $k+1$ . The handshaking procedure is spread over several TDMA frames to reduce the total network overhead. By using only one simplex transmission during each RA period, we require only one radio acquisition time and hence can shrink the length of each RA period.

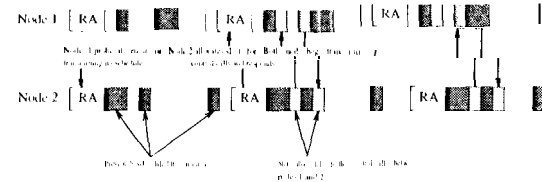


Figure 2: The handshaking procedure during random access periods.

### 3.2 Control calls

The second layer of the protocol consists of fixed-size control calls between nodes that allow a node to exchange data with its immediate (1-hop) neighbors. These control calls carry acknowledgments, local control information (requests to allocate slots for a call, notification that a particular call should be torn

down, etc.), and any information required by the routing algorithm. The slots used for the control calls on a particular link are arranged by the nodes during the contention phase as described above.

To support the power control algorithm, acknowledgements are in the form of the received SNR of the worst packet of the transmission. Thus if a particular transmission uses several slots per frame, SWAN will attempt to maintain the received SNR of the worst slot above the threshold  $\gamma$ .

A simple timeout mechanism is used to destroy links that become unusable. If a control call goes unanswered for a specified timeout period, the endpoints declare the link down and deallocate the slots that were used for all communications on that link, rerouting user calls as appropriate. Since the control calls occur during the data portion of each frame, they are subject to power control. Thus the control calls, and hence the link, will fail only under conditions not covered by the DPC-ALP algorithm (the nodes have moved so far apart that they are no longer able to communicate or mobility has made the control calls incompatible with other active calls in the network).

## 4 Example

An example of a SWAN network and schedule is given in figure 3. Recall that such a schedule is built up over time, with control calls being added as nodes agree to become neighbors. Power control guarantees that once calls are established, they are robust with respect to new calls in the network.

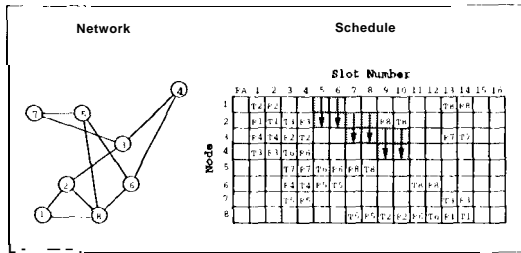


Figure 3: SWAN Network And Schedule Showing Distributed Control Information and User Call (Gray).

Idle slots are blank. Slots containing communications are marked  $XN$  where  $X$  is either  $T$  or  $R$  de-

pending on if the node is transmitting or receiving and  $N$  is the node's partner in the communication. The gray slots represent a user call from node 1 to node 4, with the arrows representing packet transmissions.

## 5 Simulation results

In this section we report the results of a computer simulation written to evaluate the SWAN protocol's performance. As described above, SWAN deals only with the formation and maintenance of individual links in the network; no node has any notion of the network topology beyond its immediate neighbors. To quantify notions such as network connectivity which deal with the existence or absence of multi-hop paths through the network, the simulations employ a simple flooding protocol which distributes network topology information to every node in the network.

We say that a path exists in the network between nodes  $i$  and  $j$  if under node  $i$ 's view of the network topology, there is a communications path from  $i$  to  $j$ . In a network of  $N$  nodes, we define node  $C_i^t$ , node  $i$ 's connectivity at time  $t$ , as the fraction of other nodes in the network to which  $i$  has a path (each node always has a path to itself). That is, if node  $i$  can form paths to  $M$  nodes in the network at time  $t$ , we say that  $i$ 's connectivity at  $t$  is  $C_i^t = M/N$ , and the network connectivity is defined as:

$$C_N^t = \sum_{i=1}^N C_i^t / N \quad (2)$$

The following parameters and assumptions were used in the simulations:

- All nodes were restricted to a 100x100 unit square. At the end of each frame, each node moved  $m$  units in a random direction. Nodes reflected off the boundaries of the square.
- There were 40 slots per frame and 5 non-interfering frequency bands. We assumed that there were a large number of orthogonal CDMA codes per frequency band and that code choice was perfect, i.e. that no two transmissions ever shared the same CDMA code.
- The required SNR was  $\gamma = 10$  and we assumed a coding gain of 20.

- Nodes were allowed to transmit at most 60 units, and at most 55 units when forming connections. That is, in the absence of interference from other nodes, a node was allowed to use a maximum transmit power such that a node 60 units away would receive the signal with  $\text{SNR} = \gamma = 10$ . To combat the formation of tenuous links, a similar strategy was used to set nodes' maximum powers when forming new connections to 55 units.
- control calls were assumed to occupy one slot, and the contention period occupied one slot during each frame.
- The timeout for unacknowledged transmissions was 25 frames for regular transmissions. Transmissions were allowed 35 frames when first established.
- Each node was allowed to maintain at most 6 neighbors.
- Each node tried to maintain three link-disjoint paths to every other node in the network. Thus when selecting nodes to which directed random access transmissions would be posted, nodes chose targets to maintain this level of connectivity first and, if there were multiple choices which yielded the same connectivity, selected the one with the highest ratio of received SNR to transmitted power.

Results of the simulation were used to evaluate the following performance measures of the SWAN protocol.

- Time to construct the network. This was defined as the minimum  $t$  such that  $C_N^t = 1$ . This is important because SWAN's contention-based incremental network formation differs greatly from the "all-at-once" formation of LCA and other previous approaches.
- Time required to add a new node to an existing network. Here the simulation was run until the network became connected. Then 100 frames were simulated to allow for conditions to settle before another node was added. The simulation was then run until  $C_N^t$  was again 1. Figure 4

shows the times required to form a network and to add a new node to an existing network, averaged over 500 simulation runs.

- The ability of the network to cope with node mobility. To study this, we recorded the connectivity of a network running the SWAN protocol over a set of 100 sample paths. The results are shown in figure 5.

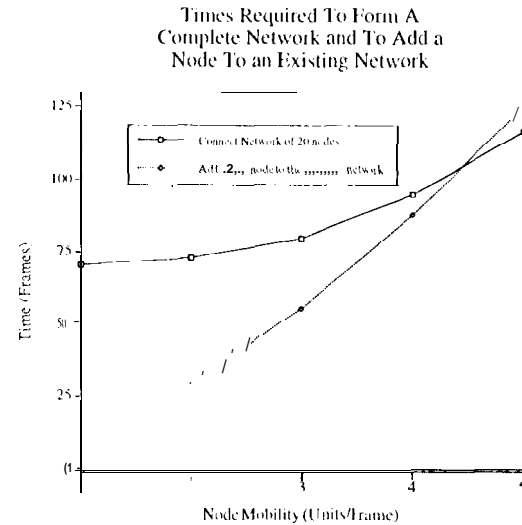


Figure 4: Time required to initially form a network of 10 nodes using 5 channels.

Note that the time required to connect a network of 20 nodes is far less than 20 times that required to add a node to an existing network. This is due to frequency reuse during link formation.

## 6 Conclusion and future work

This paper presented the SWAN protocol, a novel approach to self-organizing network management. Under the SWAN protocol, the wireless network topology is maintained in a distributed manner by using short contention periods at the beginnings of each TDMA frame, where nodes "probe" their surroundings looking for new neighbors. This method is bandwidth efficient, scalable, and provides an integrated method for handling network formation, topology maintenance, and the addition of new users. The ability of the SWAN protocol to cope with node mobility

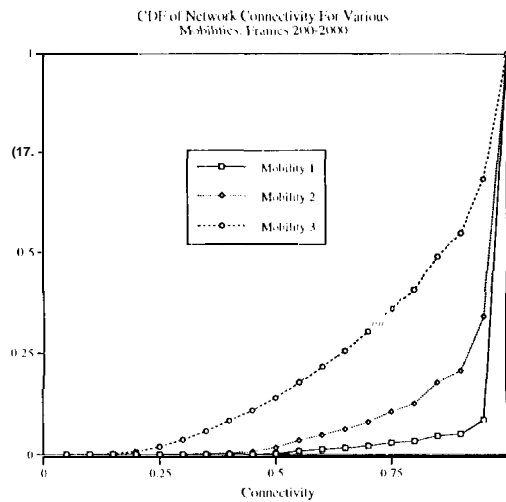


Figure 5: CDF of the network connectivity during frames 200-2000 for various values of node mobility.

compares favorably to traditional clustering methods, as shown by simulation.

We are also investigating alternate routing techniques whose main muting criterion is power conservation instead of the bandwidth/delay emphasis of the current SWAN muter. Though not suited to multi media traffic, routing to minimize power consumption is useful in distributed sensor networks where extending the operational lifetime of the system is important.

## 7 Acknowledgements

The bulk of this work was performed while the authors were at UCLA under ARPA contract ARPA-C3T0-93-112 through the WAMIS (Wireless Adaptive Mobile information Systems) project and by the National Young investigator Award NSY-N-R-9258807. Keith Scott has since joined the Jet propulsion laboratory in Pasadena, and Nicholas Bambos is now a professor at Stanford University.

## References

- [BR 90] A. Bhatnagar and T.G. Robertazzi. Layer net: a new self-organizing network protocol. in *MIL-COM 90. A New Era. 1990 IEEE Military Communications Conference.*, pages 845–849, 1990.
- [CBP94] Shou C. Chen, Nicholas Bambos, and Gregory J. Pottie. On distributed power control for radio networks. In *ICC '94*, 1994.
- [FSM89] M. Frankel, N. Shacham, and J.E. Mathis. Self-organizing networks. In *Frontiers in Computing. Proceedings of the International Conference*, pages 165–185, 1989.
- [GT95] M. Gerla and J. Tsai. Multiclusster, mobile, multi media radio network. *Wireless Networks*, 1(3):255–265, 1995.
- [LNT87] B.M. Leiner, D.L. Nielson, and F.A. Tobagi. Issues in packet radio network design. *Proceedings of the IEEE*, 75(1):6–20, January 1987.
- [PKS85] M.J. Post, A.S. Kershenbaum, and P.E. Sarachik. A distributed evolutionary algorithm for reorganizing network communications. in *MILCOM 1985*, 1985.
- [RS86] T.G. Robertazzi and P.E. Sarachik. Self-organizing communication networks. *IEEE Communications Magazine*, 24(1):28–33, 1986.
- [s1195] Keith Scott and Nicholas Bambos. A fully distributed algorithm for the creation and maintenance of wireless networks of mobile users. Technical Report UCLA-ENG-95-112, Department of Electrical Engineering, University of California at Los Angeles, 1995.
- [Tob87] F. Tobagi. Modeling and performance analysis of multihop packet radio networks. *Proceedings of the IEEE*, 75(1): 135–155, January 1987.
- [111{81}] Dennis J. Baker and Anthony Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11): 1694–1701, 1981.

## SWAN

Keith Scott

Keith.Scott@jpl.nasa.gov  
Jet Propulsion Laboratory  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099

Nicholas Bambos

bambos@leland.stanford.edu  
Department of ???  
Stanford University  
Stanford, CA

### Abstract

*There are numerous military, commercial, and scientific applications for mobile wireless networks which are able to self-organize without recourse to any pre-existing infrastructure. We present the Self-Organizing Wireless Adaptive Network (SWAN) protocol, a distributed networking protocol capable of managing such networks. The SWAN approach is based on **dynamic topology management with power control**, allowing it to adapt gradually to the changing environment instead of periodically discarding the network topology information and rebuilding the network from scratch. In addition, under SWAN **control information is distributed** instead of being concentrated in a "control phase". This provides significant savings when the acquisition times of the modems are high.*

### 1 Introduction

Given a set of mobile nodes or terminals equipped with radio transceivers, we would like for the nodes to be able to "self-organize" into a wireless network. That is, the nodes must cooperate to form a multi-hop network amongst themselves without recourse to any pre-existing infrastructure. Since the nodes are mobile, the network must continually adapt to their changing positions.

A second major challenge in such a network is to ensure that the transmission schedules of the various nodes are compatible. If a pair of nodes is using a particular transmission resource, whether it is a narrow-band frequency slot, a time slot, or a spread-spectrum

code, that resource will be unavailable to other nearby users. For example, if the nodes in a wireless network are using direct sequence spread spectrum radios to communicate, there is the danger that one transmitter may completely dominate several receivers in his vicinity, preventing them from receiving from any other node in the network (the near-far problem).

The rest of this paper is organized as follows: XXX  
YYYYZZ.

### 2 Background

#### 2.1 Other ad-hoc network methods

Several authors have examined protocols for self-organizing networks [FSM89, B&S I, PKS85, BR90, RS86, GT'95]. A common approach to maintaining the network in the face of node motility and changing radio conditions is to periodically tear down the entire network structure and to regenerate the network topology from scratch. These methods continually cycle through alternating phases of network operation. In the first phase, the protocols gather information about the network topology. This information is then used to form a compatible transmission schedule for the second phase.

This cyclic approach to network management has two main disadvantages. First, forcing all of the nodes to periodically participate in the topology gathering phase induces a large and possibly unnecessary amount of overhead. This becomes particularly difficult as the acquisition time (the time required for a receiver to "block on" to a transmitter's signal for the first time) increases. Second, generating a compatible

transmission schedule is a complex task, especially as the network size grows.

## 2.2 Distributed Power Control

SWAN uses the Distributed Power Control with Active Link Protection (DPC-ALP) power control algorithm [CBP94] to ensure that ongoing transmissions are not accidentally destroyed by new ones. Under DPC-ALP, transmitter powers are updated in a series of steps, and every transmission is in one of two states, active or inactive. The transmitter power at step  $i+1$  is a function of the transmit power in step  $i$ , the transmission state at step  $i$ , the desired signal to noise ratio  $\gamma$ , and the received SNR at step  $i$  (fed back from the receiver to the transmitter). Thus if there are  $N$  transmissions, with the power of the  $i^{th}$  transmission during the  $k^{th}$  step given by  $P_i^k$  then:

$$P_i^{k+1} = \begin{cases} \delta P_i^k & \text{Inactive Transmissions} \\ \delta P_i^k \gamma / \text{SNR}_i^k & \text{Active Transmissions} \end{cases} \quad 1 < i < N \quad (1)$$

All transmissions begin in the inactive state at a very low power (possibly commensurate with the noise power as seen by a typical receiver). A transmission becomes active once its received SNR crosses the threshold  $\gamma$ . A consequence of DPC-ALP is that, in a static network, if a transmission becomes active at step  $k$ , it will remain active for all  $t > k$  regardless of the number and locations of new transmissions started after time  $k$ . This is because active transmissions are more aggressive in updating their powers, and may in fact prevent other inactive transmissions from becoming active.

## 3 The SWAN protocol

As with many other ad-hoc networking methods, SWAN divides time into a repeating series of frames, which are further subdivided into slots, as shown in figure 1. SWAN employs the DPC-ALP power control algorithm during most of these slots to ensure that ongoing transmissions are not interrupted by transmissions being set up. In addition, there is a single slot at the beginning of each frame (possibly of a different size) that is not subject to any power control restrictions. This random access period is used to

allow nodes to form new connections within the network. Once a connection has been established between two nodes, they establish control calls to exchange the data needed to maintain the power control algorithm. These control calls form a natural means for distributing information such as routing tables.

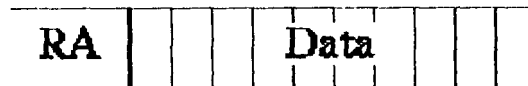


Figure 1: The TDMA frame, divided into a short random-access period and a data subframe.

### 3.1.1 Link formation and topology maintenance

To form new links in the network, nodes handshake during the random access periods at the beginning of each "TDMA frame." Each stage of the handshaking takes place during a *different* frame, so that if node  $i$  looks for neighbors during frame  $k$ , a node may respond to him during the RA period in frame  $k+1$ . The handshaking procedure is spread over several TDMA frames to reduce the total network overhead. By using only one simplex transmission during each RA period, we require only one radio acquisition time and hence can shrink the length of each RA period.

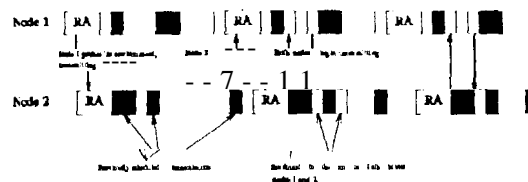


Figure 2: The handshaking procedure during random access periods.

### 3.2 Control calls

The second layer of the protocol consists of fixed-size control calls between nodes that allow a node to exchange data with its immediate (1-hop) neighbors. These control calls carry acknowledgments, local control information (requests to allocate slots for a call, notification that a particular call should be torn

down, etc.), and any information required by the routing algorithm. The slots used for the control calls on a particular link are arranged by the nodes during the contention phase as described above.

To support the power control algorithm, acknowledgements are in the form of the received SNR of the worst packet of the transmission. Thus if a particular transmission uses several slots per frame, SWAN will attempt to maintain the received SNR of the worst slot above the threshold  $\gamma$ .

A simple timeout mechanism is used to destroy links that become unusable. If a control call goes unanswered for a specified timeout period, the endpoints declare the link down and deallocate the slots that were used for all communications on that link, rerouting user calls as appropriate. Since the control calls occur during the data portion of each frame, they are subject to power control. Thus the control calls, and hence the link, will fail only under conditions not covered by the DPC-AIP algorithm (the nodes have moved so far apart that they are no longer able to communicate or mobility has made the control call incompatible with other active calls in the network).

## 4 Example

An example of a SWAN network and schedule is given in figure 3. Recall that such a schedule is built up over time, with control calls being added as nodes agree to become neighbors. Power control guarantees that once calls are established, they are robust with respect to new calls in the network.

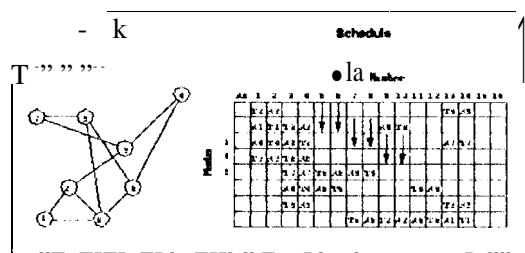


Figure 3: SWAN Network And Schedule Showing Distributed Control Information and User Call (Gray).

Idle slots are blank. Slots containing communications are marked  $XN$  where  $X$  is either  $T$  or  $R$  de-

pending on if the node is transmitting or receiving and  $N$  is the nodes partner in the communication. The gray slots represent a user call from node 1 to node 4, with the arrows representing packet transmissions.

## 5 Simulation results

In this section we report the results of a computer simulation written to evaluate the SWAN protocol's performance. As described above, SWAN deals only with the formation and maintenance of individual links in the network; no node has any notion of the network topology beyond its immediate neighbors. To quantify notions such as network connectivity which deal with the existence or absence of multi-hop paths through the network, the simulations employ a simple flooding protocol which distributes network topology information to every node in the network.

We say that a path exists in the network between nodes  $i$  and  $j$  if under node  $i$ 's view of the network topology, there is a communications path from  $i$  to  $j$ . In a network of  $N$  nodes, we define node  $C_i^t$ , node  $i$ 's connectivity at time  $t$ , as the fraction of other nodes in the network to which  $i$  has a path (each node always has a path to itself). That is, if node  $i$  can form paths to  $M$  nodes in the network at time  $t$ , we say that  $i$ 's connectivity at  $t$  is  $C_i^t = M/N$ , and the network connectivity is defined as:

$$C_N^t = \sum_{i=1}^N C_i^t / N \quad (2)$$

The following parameters and assumptions were used in the simulations:

- All nodes were restricted to a 100x100 unit square. At the end of each frame, each node moved  $m$  units in a random direction. Nodes reflected off the boundaries of the square.
- There were 40 slots per frame and 5 non-interfering frequency bands. We assumed that there were a large number of orthogonal CDMA codes per frequency band and that code choice was perfect, i.e. that no two transmissions ever shared the same CDMA code.
- The required SNR was  $\gamma = 10$  and we assumed a coding gain of 20.

- Nodes were allowed to transmit at most 60 units, and at most 55 units when forming connections. That is, in the absence of interference from other nodes, a node was allowed to use a maximum transmit power such that another node 60 units away would receive the signal with  $SNR = \gamma = 1$ . To combat the formation of tenuous links, a similar strategy was used to set nodes' maximum powers when forming new connections to 55 units.

- Control calls were assumed to occupy one slot, and the contention period occupied one slot during each frame.

- The timeout for unacknowledged transmissions was 25 frames for regular transmissions. Transmissions were allowed 35 frames when first established.

- Each node was allowed to maintain at most 6 neighbors.

- Each node tried to maintain three link-disjoint paths to every other node in the network. Thus when selecting nodes to which directed random access transmissions would be posted, nodes chose targets to maintain this level of connectivity first and, if there were multiple choices which yielded the same connectivity, selected the one with the highest ratio of received SNR to transmitted power.

Results of the simulation were used to evaluate the following performance measures of the SWAN protocol.

- Time to construct the network. This was defined as the minimum  $t$  such that  $C_N^t = 1$ . This is important because SWAN's contention-based incremental network formation differs greatly from the "all-at-once" formation of FCA and other previous approaches.
- Time required to add a new node to an existing network. Here the simulation was run until the network became connected. Then 100 frames were simulated to allow for conditions to settle before another node was added. The simulation was then run until  $C_N^t$  was again 1. Figure 4

shows the times required to form a network and to add a new node to an existing network, averaged over 500 simulation runs.

- The ability of the network to cope with node mobility. To study this, we recorded the connectivity of a network running the SWAN protocol over a set of 100 sample paths. The results are shown in figure 5.

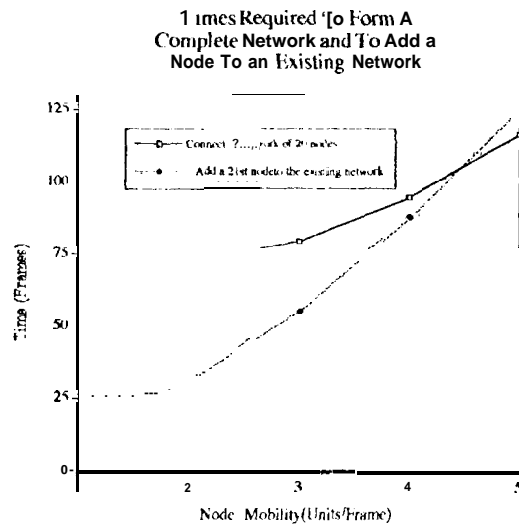


Figure 4: Time required to initially form a network of 10 nodes using 5 channels

Note that the time required to connect a network of 20 nodes is far less than 20 times that required to add a node to an existing network. This is due to frequency reuse during link formation.

## 6 Conclusion and future work

This paper presented the **SWAN** protocol, a novel approach to self-organizing network management. Under the SWAN protocol, the wireless network topology is maintained in a distributed manner by using short contention periods at the beginnings of each TDMA frame, where nodes "probe" their surroundings looking for new neighbors. This method is bandwidth efficient, scalable, and provides an integrated method for handling network formation, topology maintenance, and the addition of new users. The ability of the SWAN protocol to cope with node mobility

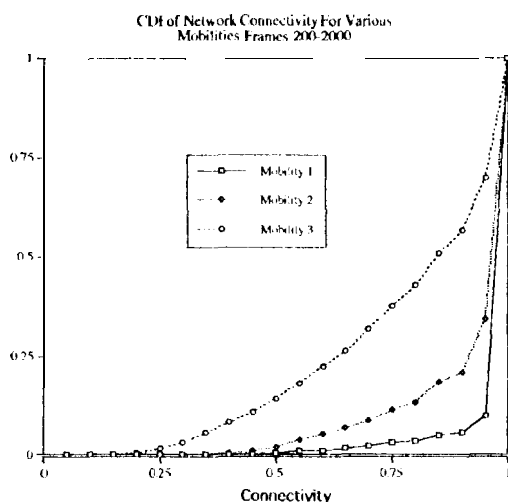


Figure 5: CDF of the network connectivity during frames 200-2000 for various values of node mobility.

compares favorably to traditional clustering methods, as shown by simulation.

We are also investigating alternate routing techniques whose main routing criterion is power conservation instead of the bandwidth/delay emphasis of the current SWAN router. Though not suited to multimedia traffic, routing to minimize power consumption is useful in distributed sensor networks where extending the operational lifetime of the system is important.

## References

- [BL81] Dennis J. Baker and Anthony Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11): 1694-1701, 1981.
- [BR90] A. Bhatnagar and T.G. Robertazzi. Layered: a new self-organizing network protocol. In *MILCOM 90. A New Era. 1990 IEEE Military Communications Conference*, pages 845-849, 1990.
- [CBP94] Shou C. Chen, Nicholas Bambos, and Gregory J. Pottie. On distributed power control for radio networks. In *ICC '94*, 1994.
- [FSM89] M. Frankel, N. Shacham, and J.E. Mathis. Self-organizing networks. In *Frontiers in Computing. Proceedings of the International Conference*, pages 165-185, 1989.
- [GT95] M. Gerla and J. Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 1(3):255-265, 1995.
- [LN187] B.M. Leiner, D.L. Nielson, and F.A. Tobagi. Issues in packet radio network design. *Proceedings of the IEEE*, 75(1):6-20, January 1987.
- [PKS85] M.J. Post, A.S. Kershbaum, and P.E. Sarachik. A distributed evolutionary algorithm for reorganizing network communications. In *MFCOM 1985*, 1985.
- [RS86] T.G. Robertazzi and P.E. Sarachik. Self-organizing communication networks. *IEEE Communication Magazine*, 24(1): 28-33, 1986.
- [SB95] Keith Scott and Nicholas Bambos. A fully distributed algorithm for the creation and maintenance of wireless networks of mobile users. Technical Report UCLA-ENG-95-112, Department of Electrical Engineering, University of California at Los Angeles, 1995.
- [Tob87] F. Tobagi. Modeling and performance analysis of multihop packet radio networks. *Proceedings of the IEEE*, 75(1):135-155, January 1987.